AD-A233 601

CUMENTATION PAGE

Form Approved OMB No 0704 0188

1a REPORT SECURITY CLASSIFICATION: Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: The Regents of the University of California	6b OFFICE SYMBOL (If applicable)
7a NAME OF MONITORING ORGANIZATION: Cognitive Science Program, Office of Naval Research (Code 1142PT)	
6c ADDRESS (City, State, and ZIP Code): University of California, Los Angeles, Office of Contracts and Grants Administration, Los Angeles, California 90024	7b ADDRESS (City, State, and ZIP Code): 800 North Quincy Street, Arlington, VA 22217-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION: Defense Advanced Research Projects Agency	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-86-K-0395	
8c ADDRESS (City, State, and ZIP Code): 1400 Wilson Boulevard, Arlington, VA 22209-2308	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61153N	RR04206	RR04206-0C	442c022

11 TITLE (Include Security Classification): Human Benchmarking Methodology for Expert Systems

12 PERSONAL AUTHOR(S): O'Neil, Harold F., Jr.; Li, Yujing; Jacoby, Anat, & Swigger, Kathleen M.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Interim	FROM 7/1/89 TO 9/30/90	September 1990	34

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial intelligence, expert systems, human benchmarking
12	05		

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This document outlines the strategy used for benchmarking expert systems to human performance. Two major alternatives for human benchmarking of expert system are proposed: computer science driven or psychological process driven. The computer science driven approach is either (1) expert system driven in which one picks an expert system which encodes an expert, derives psychological processes, and tests the processes with people, or (2) expert system shell driven in which one estimates the "intelligence" of the shell (parent), assumes that applied expert systems will have similar "intelligence," then follows the procedures of the expert system driven approach. The psychological process driven approach involves picking a psychological process, finding an expert system that exemplifies the process, and then tailoring a test for the expert system and for people. The computer science driven approach was used to develop the expert system benchmarking methodology by relating cognitive taxonomies and expert system taxonomies.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL: Dr. Susan Chipman	22b TELEPHONE (Include Area Code): (703) 696-4318	22c OFFICE SYMBOL: ONR 1142CS

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

Project Report #20

HUMAN BENCHMARKING METHODOLOGY FOR EXPERT SYSTEMS

Harold F. O'Neil, Jr.

Cognitive Science Laboratory
University of Southern California

Yujing Ni

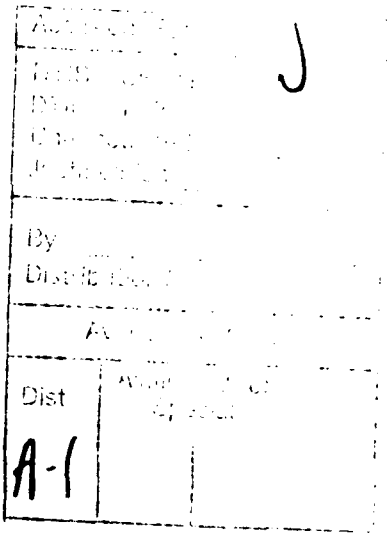
Anat Jacoby

Center for Technology Assessment
UCLA Center for the Study of Evaluation

Kathleen M. Swigger

Department of Computer Sciences
University of North Texas

September 1990



Artificial Intelligence Measurement System
Contract Number N00014-86-K-0395

Principal Investigator: Eva L. Baker

Center for Technology Assessment
UCLA Center for the Study of Evaluation

This research report was supported by contract number N00014-86-K-0395 from the Defense Advanced Research Projects Agency (DARPA), administered by the Office of Naval Research (ONR), to the UCLA Center for the Study of Evaluation. However, the opinions expressed do not necessarily reflect the positions of DARPA or ONR, and no official endorsement by either organization should be inferred. Reproduction in whole or part is permitted for any purpose of the United States Government.

The authors wish to thank other member of the UCLA human benchmarking group: Robert Brazile, Frances Butler, and Merlin Wittrock.

Background

The Cognitive Science Laboratory of USC has a subcontract with the Center for Technology Assessment of the UCLA Center for the Study of Evaluation to assist in the evaluation of expert systems using a human benchmarking methodology. In turn, UCLA has an existing contract with the Defense Advance Research Projects Agency to study methodologies for the evaluation of artificial intelligence systems. UCLA systems of interest in artificial intelligence have included vision, natural language, expert system shells, and expert systems. The purpose of the current subcontract is to assist in the human benchmarking of experts systems.

Benchmarking is defined by Guralnik (1984, p.131) in the following manner: "(1) Surveyor's mark made on a permanent landmark of known position and altitude, used as a reference point in determining other attitudes, (2) standard or point of reference in measuring or judging quality." Benchmarking is also used as a term to denote a standard process for measuring the performance capabilities of software and hardware systems from various vendors (Bentwell, 1974; Letmanyi, 1984). It is the latter sense that we use benchmarking: Thus, human benchmarking is an evaluation procedure by which an AI system's performance is judged based on a sample of people's performance on tasks with psychological fidelity.

There are two major human benchmarking alternatives: computer science driven or psychological process driven. The computer science driven approach is either (1) expert system driven in which one picks an expert system which encodes an expert, derives psychological processes, and tests the processes with people; or (2) expert system shell driven in which one estimates the "intelligence" of the shell (parent), assumes that applied expert systems will have similar "intelligence," then follows the procedures of the expert system driven approach.

The second alternative for human benchmarking is psychological process driven in that one picks a psychological process, finds an expert system that exemplifies the process, and then tailors a test for the expert system and a test using people. For us, the psychological process approach was not feasible due to a constraint of available AI developers with robust programs who were willing to collaborate with us. Thus, we chose the computer science driven approach. In turn, because our effort is focused on expert systems we chose this sub-approach rather than focusing on shells.

In order to provide an intellectual foundation for human benchmarking, a literature review was conducted and documented as a prior deliverable (O'Neil, Ni, & Jacoby, 1990). The literature was reviewed from two viewpoints: (1) a computer science and software engineering perspective and (2) a cognitive science perspective with a focus on psychological assessment.

This literature review suggests that there are different approaches to expert system evaluation including evaluation criteria and evaluation procedures. The literature offered diverse environments to capture developmental aspects of expert systems evaluation. The review suggests the possibility of developing a psychometric standard for the evaluation of expert systems; the review also helped us to document similarities and differences between cognitive psychology and artificial intelligence, which is important for our human benchmarking approach. Further, it suggested multiple frameworks useful for assessment.

For example, O'Keefe, Balci, and Smith's (1987) methods of validation of an expert system attempt to capture the whole developmental process for expert systems (see Table 1). They categorized validation methods as qualitative and quantitative. These validation methods are described in Table 1. In Table 1, all methods are qualitative except the last. The face validation and the predictive validation methods are preliminary approaches to validation during the development of the system. Turing tests and field tests

are validation methods used after the installation of the system. Others are validation methods applicable both during and after the implementation of the expert system. Many of these techniques depend upon test cases.

Table 1
Methods for Validation Evaluation
(Adapted from O'Keefe et al., 1987)

Method	Description
Face validation	Preliminary comparison of system performance with expert performance against a prescribed performance range
Predictive validation	Assessment of performance by using historic cases and either (1) known results or (2) measures of human expert performance on these cases
Turing test	Expert judges' blind evaluation of both system and human expert performance for given cases
Field test	Evaluation of prototypical expert system in the intended context
Subsystem validation	Decomposition of subsystems in an expert system and evaluation of the performance of each subsystem under given input data
Sensitivity analysis	Validation of system by systematically changing expert system input variables and parameters over some range of interest and observing the effect on system performance
Visual interaction	A validation environment in which the experts' direct interaction with an expert system allows for face validation, sub-system validation, and sensitivity validation
Quantitative validation	Statistical techniques to compare expert system performance against either test cases or human experts

However, the test cases in many studies cited in the literature either were generated by an expert or experts in correspondence with his or her perceived pre-specification for a system or previously used examples in evaluation of similar systems. The representativeness of the cases seems to be problematic because the test-case generation technique relies only on the experts' arbitrary judgments which vary across situations and over time. In fact, the representativeness issue occurs during the process of expert system development. The domain knowledge coded in an expert system is usually extracted from one or two experts in the field. However, there is great variation in domain experts for a given problem. Even the same expert may change his solution methods for the same problem over time. Thus, using only prespecifications as evaluation criteria and a very small sample of test cases make the representativeness of the test cases uncertain. The lack of representation would, of course, reduce the validity of the methods specified by O'Keefe et al. (1987).

UCLA/USC personnel have been engaged in developing a model for the evaluation of expert systems for the past year. This ongoing project has resulted in a detailed characterization of the development process in terms of stages of development, evaluation considerations, and knowledge engineer question types throughout the process. Documentation has been compiled in a case study methodology on some differences between traditional software and artificial intelligence systems in the development process with changes in project organization, size, and purpose; application type; knowledge of programming environment; and personnel attributes. These differences have implications for the complexity of a development model as well as the kind, purpose, and timing of measurements that might be made during the development (Slawson 1987; Slawson, Hambleton, & Novak, 1988).

Our human benchmarking approach is to establish an evaluation, that is, to norm an expert system's performance on a sample of people's performance. The implication of this approach is that it goes beyond the conventional approach of expert system evaluation and aims to build psychometric criteria through comparison of an expert system's performance with differentiated performances by people.

The basic idea of a human benchmarking approach is to use people's performance(s) to evaluate an artificial intelligence system. Following this line of thought, there are three methods for the approach. They are (1) the Turing test, (2) comparison of the system's performance with persons' performance of different expertise in the same field, and (3) evaluation of the system's performance against a norm generated from the assessment of performance by groups from both inside and outside the relevant field. These methods differ in their samples of people and their evaluation criteria. The sample of people in the Turing test is only the expert group; in the second method, the sample includes people with different levels of expertise in the same field. In the third method, both experts and nonexperts comprise the sample from both inside and outside the relevant field. In terms of evaluation criteria, the first two methods usually use prespecifications for a system as evaluation criteria, while the third uses psychometric-oriented criteria generated from test specifications based on a structural analysis of a domain task. Due to the features in the sample of people and the evaluation criteria, the third method appears to have a potential generalizibility which may lead to directly evaluating a system's "intelligence" and carrying out comparisons among similar or even different kinds of systems. It is this methodology that we are attempting to explore.

The function of norms is to help establish the status of the expert system performance. For example, monitoring is a conscious, cognitive function which is directed at the acquisition of information about the status of ongoing problem-solving processes. If

0% of 4th graders can monitor and 100% of undergraduate students can monitor effectively, then an expert system with monitoring as its major function exhibits the "intelligence" of no greater than an undergraduate student. Subsequent research would investigate 5th graders through college seniors to get a range of equivalent intellectual functioning by the program. Norms are differential estimates in the research literature. For example, on the Stanford-Binet intelligence test, norms are percentages passing a task by grade level. With standardized tests, the process varies. For instance, raw scores on the Iowa Test of Basic Skills are converted to either developmental scores (grade equivalent, age equivalent, and standard scores) or status scores (e.g., percentile ranks) (Hieronymous, Hoover, & Lindquist, 1986).

Finally, one could evaluate the performance of a system against human developmental data. For example, the data from developmental psychology suggests that classification in terms of function of objects is viewed as a higher level than classification in terms of the objects' shape or color (e.g., Bruner, Olver, & Greenfield, 1966). For example, young children view three black cats similarly merely because all of them are black. Older children, on the other hand, treat them as a group because they belong to the same kind of animal, i.e., cats. Thus, the same judgment may be made in expert system applications. It may be appropriate to say that a system doing function classification is "smarter" than that doing shape or color classification although both carry out the same classification function in a technical sense.

While investigating the current state of the art in evaluation of expert systems, we located a large number of studies that either dealt with expert system evaluation or included evaluation as one component of expert system development. The literature (O'Neil et al., 1990) also included a subset of the following domains: (a) metacognitive skills (e.g., monitoring skills); (b) qualitative and quantitative methodologies; (c) software engineering; and (d) intelligent tutoring systems. Based on this literature review, the UCLA/USC team

is focusing on the modification of the "human benchmarking" methodology from the natural language area and applying it to the area of expert systems.

Modification of Human Benchmarking Methodology for Expert Systems

A human benchmarking approach was initially applied to a natural language understanding system (Baker & Lindheim, 1988; Baker, Turner, & Butler, 1990). To extend it to the evaluation of expert systems, we modified this method by relating cognitive psychological taxonomies to an expert system taxonomy (e.g., monitoring). With this bridge, we developed measures for groups of people to allow us to benchmark an expert system.

Natural Language Application of Human Benchmarking

The human benchmarking method has been preliminarily explored in the evaluation of a natural language system (Baker & Lindheim 1988; Baker et al., 1990). The authors defined human benchmarking as assessing an artificial intelligence system in terms of similar performance by groups of people.

A natural language understanding system, IRUS, was selected as the target for the human benchmarking approach. IRUS is an interface which has natural language facility and allows a user to access a database through a natural language by asking questions. A sample of IRUS queries was collected from a list of 163 queries that were used in an August 1986 Navy demonstration of the capabilities of IRUS (Baker et al., 1990). A linguistic analysis of these queries served as the basis of test specifications for the development of the Natural Language Elementary Test (NLET). The NLET test was

designed to duplicate the queries that the system, IRUS, was able to understand. Sentence types IRUS was able to handle became the test specifications for NLET. By mediation of the test specifications, the NLET test is "equivalent" to IRUS in linguistic functioning (see Table 2).

Table 2

Example NLET Form 3 Items and Parallel IRUS Queries by Item Type

(Adapted from Baker et al., 1990)

IRUS Query	Linguistic Structure	NLET Item
Does FREDERICK have TASCOM?	Noun Phrase (NP) + Verb + NP	Does the dog have a hat?
Display all carriers in the PACFLT.	Performance Verb + NP	Point to the car with a flag.
What's HAMMOND's readiness?	NP + Copula + NP	Who is the driver of the truck?
How many cruisers are in WESTPAC?	NP + Verb or Copula + Prepositional Phrase	How many striped snakes are on the floor?
List the ships that are C4 or that are C5.	Performance Verb + NP + Relative Clause	Point to the cars that have a flag or that have stripes.
Are there any submarines in the South China Sea?	Non-referential "there" + Copula + NP	Are there any snakes in the house that are striped?

The NLET test has 39 items of these linguistic structures. The reliability of the test among its items has not been established because there are several sets of items that are constructed depending on preceding items. The NLET test was administered to two groups of students including kindergarteners and first graders. In principle, IRUS would be rated as first grade level of natural language understanding in a syntactic sense if 90% of first graders could understand all or most of NLET items. However, the human benchmarking approach was designed to "norm" performance of AI systems with differentiated performances by groups of people. Therefore, a standardized test was needed so that a score of NLET could be interpreted or equated in a score of a standardized test. Then, it would be possible to "norm" performance of AI systems. Thus, the language section of the Iowa Test of Basic Skills (ITBS), a standardized test, was used in the human benchmarking study.

It was expected that the first graders would perform better than the kindergarteners on both NLET and ITBS tests. This continuation then could be used as a "scale" to measure performance of IRUS. But the results showed that these two groups' performance was similar on both tests. However, there was a developmental trend in their NLET performance when the subjects were grouped according to their grade equivalent scores on the ITBS standard language ability test (see Table 3). This suggested that the human benchmarking approach could be further specified with more groups of students.

Table 3
NLET Test Descriptive Statistics for ITBS Grade-Equivalent Bands
(Baker et al., 1990)

ITBS Grade Equivalent	N	NLET Mean	NLET SD	NLET Range
3.4-4.9	17	34.47	2.27	31-38
2.0-2.9	20	32.85	1.57	29-35
1.0-1.9	37	32.59	1.97	22-37
K.0-K.9	42	32.00	2.95	26-37
P.0-P.9	10	30.60	3.20	26-35

The natural language human benchmarking methodology relies heavily on domain referenced testing as its central metaphor. The equivalence between IRUS queries and NLET items was determined in terms of syntactic structures alone.

However, one expects that human benchmarking of expert systems would be different than human benchmarking of natural language understanding systems. First, an

expert system always involves a considerable amount of domain-specific knowledge; thus, unlike natural language systems, it is difficult to isolate the structure of a task from its content for test specifications. For example, assigning airline flights to airport gates needs specific strategies as well as knowledge about the airport. In addition to capturing the requirements of domain-specific knowledge for a given task, one needs to take into consideration measurement of relevant experiences in human benchmarking of expert systems. For example, travel experience and scheduling experience would be relevant if one is to assign airline flights to airport gates.

Thus, as expected, we modified the methodology designed for an NL application by augmenting it with metaphors and methodologies from the metacognitive skills literature. The basic idea was to map the methodologies and measures of human monitoring skills (e.g., Beyer, 1988; Weinstein & Mayer, 1986) onto expert systems in the area of scheduling. In order to achieve this goal we required a measuring instrument to measure human cognitive skills, something analogous to NLET in the natural language area. Our preference was to select a reliable and valid commercially available instrument from the literature. Thus, our literature review (O'Neil et al., 1990) also focused on cognitive skills instruments.

Explicit Measurement of Cognitive Skills

The search for cognitive skills instruments was intended to assist us in the measurement aspects of a "human benchmarking" approach to the evaluation of expert systems. The work began by investigating several cognitive and metacognitive skills, that is, monitoring, problem solving, reasoning, inference, planning, diagnosis, and scheduling. The selection reflected the combined perspectives from cognitive psychology and expert system applications. For example, "problem solving," "reasoning," "inference," "monitoring" and "planning" were considered because they are almost the

same labels in expert system applications and cognitive psychology, whereas "diagnosis" and "scheduling" are from categories that are unique to expert system applications.

Monitoring. As a general human executive process, monitoring is directed at the acquisition of information about and the regulation of one's own ongoing problem solving processes, which is described as "cognitive monitoring" by Flavell (1981), "metacognition" by Sternberg (1985), and "executive decision" by Kluwe (1987). Thus, one monitors the internal world (i.e., one's thoughts). However, monitoring as an executive process can also use information from monitoring one's thoughts of the external world (e.g., a display).

One possible positive effect of the monitoring function is the ease with which cognitive strategies are transferred to new task demands (Kluwe, 1987). However, frequent monitoring does not necessarily lead to successful performance (Kluwe, 1987). Successful performance is not determined only by the monitoring function: there are interactions between the monitoring function and other cognitive skills as well as prior experience and knowledge. For example, effectiveness of monitoring may be influenced by task proficiency. Hickman (1977) interviewed two widely read professional persons, asking them to reflect upon their own comprehension processes while reading. Their comments demonstrated a clear sense of purpose for reading, a very active use of identifiable strategies, and an emphasis on relating prior experience and knowledge to material read. Unfortunately, we were not able to find a commercially available instrument for the measurement of monitoring.

Planning. Planning, in cognitive psychology, is viewed as a general cognitive skill or as a context-specific activity (Scholnick & Friedman, 1987). As a general cognitive skill, it is defined as the predetermination of a course of action aimed at achieving a goal (Hayes-Roth & Hayes-Roth, 1979); it is also considered as a series of subfunctions of

metacognition, e.g., stating a goal by predicting results (Beyer, 1988). As a context-specific activity, it emphasizes context-specific strategies such as reading comprehension, running a series of errands, etc. (e.g., Jacobs & Paris, 1987; Boynton, 1986). Our literature review indicated no commercially available instruments to measure planning.

Diagnosis and Scheduling. As mentioned above, diagnosis and scheduling are from the categories of expert system applications. There are no such categories in cognitive skills and thus no measures.

Table 4 summarizes the concepts from an artificial intelligence and cognitive psychology perspective. After an extensive literature review we found no standardized measurement instrument of any construct in Table 4.

Table 4

Monitoring, Planning and Scheduling

AI	Psychology
Monitoring	**Monitoring**
Monitoring observed behavior of a system to examine whether the function of a system is deviant from expected behavior. (Waterman, 1986)	As a cognitive executive process directing at monitoring (e.g., self checking one's own ongoing cognitive process. (Sternberg, 1985; Flavell, 1981; Kluwe, 1987)
Planning	**Planning**
A problem-solving strategy which is defined as the predetermination of a course of action aimed at achieving a goal (Hayes-Roth & Hayes-Roth, 1979)	A problem-solving strategy which is defined as the predetermination of a course of action aimed at achieving a goal (Hayes-Roth & Hayes-Roth, 1979)
Deciding on an entire course of action before acting, such as develop a plan for attacking enemy airfields (Waterman, 1986)	
Diagnosis	**Diagnosis**
The process of fault-finding in a system based on interpretation of potential indicator data (Hayes-Roth, Waterman & Lenat, 1983)	Not a specific research area.
Scheduling	**Scheduling**
Selecting a sequence of operations needed to complete a plan, determines start and end time, and assigns resources to each operation (Waterman, 1986)	Not a specific research area

Alternative Measurement of Metacognition

Although there are no commercially available standardized tests for the measurement of metacognition, cognitive researchers have designed and used various

approaches to measurement. Usually one infers the presence or absence of metacognition by some sort of experimental variation. The techniques for measurement of metacognition in empirical studies may be categorized into five kinds. The following are examples of these methods.

Error detection paradigm. This method involves constructing reading material with contradictory information. If the inconsistent information goes undetected, it is assumed that the reader has failed to monitor his or her level of comprehension adequately. This method is mostly used in the measurement of reading comprehension monitoring. A short passage is provided to a subject which contains a single contradiction. The contradictory information is usually not in contiguous sentences. For example, one passage describes cave-dwelling bats that are deaf; toward the end of the passage it is stated that the bats use echoes to locate objects (e.g., Walczyk & Hall, 1989). The subject is asked to detect the contradictions. The number of correct detections is used as an index of reading comprehension monitoring. Higher numbers indicate more monitoring.

Self-rating scale. This type of measurement involves asking participants to answer or self report on statements about metacognition. For example, the reading awareness interview was designed to assess children's awareness about reading in three areas: evaluating task difficulty and one's own abilities, planning to reach a goal, and monitoring process towards the goal. The interview contained scale items (Jacobs & Paris, 1987). For example, one monitoring item is "Why do you go back and read things over again?" with three scored choices: (a) because it is good practice (one point); (b) because you didn't understand it (two points); (c) because you forgot some words (0 points). Another example is learning strategy inventories (e.g., Weinstein & Mayer, 1986).

Think-aloud protocol analysis. Think-aloud protocol analysis is a psychological research method. It asks a subject to vocalize aloud his or her thinking

processes while he or she is working on a problem. The data as a protocol are then coded according to a specified model for psychological analysis which provides insights into the elements, patterns, and sequencing of underlying thought processes (Erission & Simon, 1980). Hayes-Roth and Hayes-Roth (1979) and Boynton (1986) used this method to study cognitive processes of planning. For example, their studies suggest that monitoring and self-checking are important components of planning.

Another example of think-aloud protocol involves subjects working in pairs to solve a problem. Their verbal statements during the problem-solving period are collected and examined for metacognition involved in the process. For example, pairs of subjects were asked to solve LOGO programming problems (Clements, 1987). While they were solving LOGO problems, their verbal statements were either tape recorded or videotaped. The experimenter would stimulate verbalization if needed. Verbatim transcripts were prepared. The subjects' recorded statements were categorized in the scheme of Sternberg's componential intelligence model (1985) such as deciding on the nature of the problem, selecting performance components, combining performance components, monitoring solutions, etc. For each subject, the number of statements in a given category divided by the total number of statements yielded a percentage of occurrence for that category. The sum of the four variables mentioned above defined the metacomponential processing score. The results showed that correlations between the metacomponential processing score and performance on error detection was positive and significant.

Evaluating relative efficacy of strategies used. This method is used to assess subjects' awareness of effectiveness of strategies used. The underlying assumption is that one aspect of metacognition is to monitor strategies for use in the regulation of ongoing cognitive activity. For example, two memory strategies were provided to subjects when they were asked to remember a list of vocabulary. Then they were asked to evaluate the relative efficacy of the strategies according to their memory experience (Brigham &

Pressley, 1988). It was assumed that awareness of task-appropriate strategies facilitates success.

Behavior observation. It is expected that changes in problem solving conditions would result in changes in cognitive processes that would be, in turn, reflected in overt behavior measures such as speed. For example, subjects were requested to solve puzzles under reversible and irreversible conditions (the irreversible condition being that once a piece of the puzzle was placed on the working cardboard it became fixed and could not be removed). Changing the problem solving condition would cause children to increase their intensity and thus decrease the speed of their solution approach. The change of actions was viewed as the evidence of monitoring their problem solving processes (Kluwe, 1987). The "data" in this case are not the subjects' cognitions but their overt external behaviors. In Kluwe's study, some children tended to increase checking behaviors such as holding in hand a piece of the puzzle longer or trying several pieces of the puzzle elsewhere rather than putting them directly on the working cardboard.

Cognitives and Expert System Taxonomies

In order to compare the functioning of both software and persons, one requires some classification (or taxonomy) to categorize particular cases as instances of more general constructs. Then one can measure comparable functioning at an appropriate abstract level. Only when a particular category or function of an expert system is identified to be parallel to a particular type of cognitive functioning, can the decision be made that the particular measuring instruments are legitimate to be used for comparing the performance of expert systems and that of human beings. Both general and specific approaches were used to accomplish this task.

Our general approach used two category systems (or classifications/taxonomies) of expert systems (Chandrasakaran, 1986; Shalin, Wisniewski, & Levi, 1988). Their possible correspondences were examined based on the descriptions or definitions of the categories or functions of expert systems and cognitive skills.

As shown in Table 5, Shalin et al.'s system (1988) categorizes expert systems according to their functions and knowledge requirements. These five categories are classification, interpretation, design as well as problem solving and planning. They are described as hierarchically inclusive because the functions and the knowledge requirements for more complex expert system functions subsume requirements for less complex expert systems. Chandrasekaran's system (1986) identifies four critical functions or features called "generic tasks." The five critical functions are hierarchical classification, hypothesis matching or assessment, abductive assembly, hierarchical design by plan selection, and state abstraction. The "generic task" analysis seems to be more able to capture similar functions from different expert systems because it focuses on general functions of expert systems rather than specific tasks they perform, which is consistent with to our approach in benchmarking expert systems.

However, the function of human cognitive monitoring and expert system monitoring may differ in that human cognitive monitoring regulates on-line cognitive processes of problem-solving, while a monitoring system does an on-line task such as weather monitoring, etc. It may be in this sense that Shalin et al. (1988) viewed the function of a monitoring system as similar to that of a classification system which functions so as to match the input features of an example of a class to a concept or its internal representation of the class. Thus, the functions of cognitive monitoring and an expert system monitoring something external are not the same thing. Although the same word is

used, "monitoring" does not mean the same thing in expert system applications and cognitive psychology.

Table 5

Expert System Function Classification

Shalin et al. (1988)

Classification
Matches the input features of an exemplar of a class to a concept

Interpretation
Construct a coherent representation from classified objects

Design
Arranges objects according to constraint on these objects

Problem-solving and planning
Arranges actions according to constraints on action sequences

Chandrasekaran (1986)

Hierachical classification
Organize concepts in terms of their relations with the top-most concept having control over the sub-concepts

Hypothesis matching or assessing
Generate a concept, match it against relevant data, and determine a degree of fit

Hierarchical design by plan selection and refinement
Choose a plan based on some specification, instantiates and executes parts of the plan, which in turn suggests further details of the design

State abstraction
Predict a state change when a proposed action may be executed

In order to be clear about the meaning of monitoring one needs to define the construct of self-monitoring in human terms. Our definition of this construct is a synthesis of Weinstein and Mayer (1986) and Bayer (1988) (see Table 6). We view self-monitoring as conscious and periodic self-checking of whether one's goal is achieved and, when necessary, selecting and applying different strategies. Thus, to self monitor one must have a goal (either assigned or self-directed) and one must have a cognitive strategy to monitor (e.g., finding the main idea). Further, one needs a mechanism to know which strategy among competing strategies to initially select to solve a task and, further, when to change such a strategy when it is ineffective in achieving the goal. A problem arises when one's initial cognitive strategy is ineffective but there is no other strategy to select. The latter case is common with lower aptitude students. For example, many low aptitude students memorize information by repetition alone and have no other strategies to use when repetition is inefficient or ineffective. Thus, although conscious of failure, they have no other strategy to select and use (e.g., use of imagery). Surprisingly, the cognitive term of self-monitoring finds its best realization in the expert system application of scheduling.

The typical goal of an expert system scheduler is to assign an item to a specific time, location, etc., without violating any constraints. The scheduler makes assignments according to some prescribed strategy such as the number of constraints assigned to the item or the distance among items. If an expert system's initial strategy fails to assign all the items, then the program must either relax its constraints or adopt a new and different strategy (Dhar & Raganathan, 1988). Similar to the human example, a more sophisticated and complex expert system should be able to select a different scheduling strategy if its initial strategy proves ineffective.

Table 6

Metacognition Taxonomy (adapted from Beyer, 1988)

Planning
--Stating a goal
--Selecting operations to perform
--Sequencing operations
--Identifying potential obstacles/errors
--Identifying ways to recover from obstacles/errors
--Predicting results desired and/or anticipated
Monitoring
--Keeping the goal in mind
--Keeping one's place in a sequence
--Knowing when a subgoal has been achieved
--Deciding when to go on to the next operation
--Selecting next appropriate operation
--Spotting errors or obstacles
--Knowing how to recover from errors, overcome obstacles
Assessing
--Assessing goal achievement
--Judging
--Evaluating appropriateness of procedures used
--Assessing handing of obstacles/errors
--Judging efficiency of the plan and its execution

Our specific approach using cognitive and expert system taxonomies investigated two software systems—GATES and ART. ART (Inference Corp., 1987) is an expert system shell. GATES (Brazile & Swigger, 1988) is an expert system designed to assign airplanes to gate at airports.

Expert system shell—ART. ART was chosen for analysis because (1) it is a versatile tool that incorporates a sophisticated programming workbench in common use; and (2) we have expertise immediately available in ART, as one member of our research group had extensive experience with this shell.

ART consists of several components: facts, schemata, viewpoints, logical dependencies, rules including forward chaining and backward chaining, object oriented programing, and graphics. The first six components were examined for their possible parallels to cognitive skills. The result of the analysis is depicted in Table 7.

As shown in Table 7, a fact in ART is conceptualized as declarative knowledge in a cognitive taxonomy, and the feature of the schemata component in ART is identical to that of the schema description in cognitive psychology (Rumelhart & Ortony, 1977). The forward chaining and backward chaining rules in ART are similar to inductive reasoning or bottom-up processing and deductive reasoning or top-down processing in cognitive skill taxonomies, respectively. However, cognitive top-down and bottom-up processing are more comprehensive than forward and backward chaining. Forward and backward chaining follow relatively fixed, step-by-step procedures, while the cognitive top-down/bottom-up approach is more reflective (e.g., Vygotsky, 1978). Viewpoint and logical dependency in ART have no parallels in cognitive psychology. Although interesting, this analysis did not lead to action as one would still have to create an expert system with the shell to provide a human benchmarking test. Thus we turned to a specific expert system, GATES.

Table 7

A Parallel Analysis of ART Components and Cognitive Skills or Functions

ART Components	Cognitive Skills or Functions
Fact an item of information	**Declarative Knowledge**
Schemata collection of facts that represents an object or class of objects that share certain properties	**Scheme** a data structure for a cluster of concepts about an objects, events, situations
Viewpoints a means of segregating data into separate models of the situation that an application is considering	**No fit**
Logical Dependencies the logic dependence of some facts on other facts in a data base	**No fit**
Forward Chaining the presence of certain facts allow reasoning to reach an appropriate conclusion	**Bottom-Up Processing** concepts activated from a lower to a higher order
Backward Chaining the ability to reason from a desired conclusion in search of the facts that might substantiate it	**Top-down Processing** concepts activated from a higher order to a lower order

Expert system—GATES. GATES is an expert system written in Prolog for gate assignment at TWA 's JFK and St. Louis airport terminals (Brazile & Swigger, 1988; 1989). It is a production program with documentation and source code available. It was chosen as a target system for our human benchmarking approach for two reasons. First, the system has some features (e.g., monitoring functions) we are interested in for the benchmarking evaluation of expert systems. Second, we have a good cooperative relationship with the developers of the system. In this section we will provide a brief

description of the system and an analysis of the correspondence between the system's functions and Shalin et al.'s (1988) and Chandrasekaran's (1986) categories.

The GATES system. GATES is a constraint satisfaction expert system developed to create TWA's monthly and daily gate assignments at Kennedy Airport. Obtained from an experienced ground controller, the domain knowledge is represented in Prolog predicates as well as several rule-like data structures including permission rules (the GATEOK predicate) and denial rules (the conflict predicate). These two kinds of rule determine when a set of gates can or cannot be assigned to a particular flight.

The system uses the following procedures to produce monthly gate assignments:

1. Considering an unassigned flight that has the most constraints first (a set of FLIGHT rules);
2. Selecting a particular gate for a particular flight by using a set of GATEOK rules that have been arranged in some priority;
3. Verifying whether the gate assignment is correct by checking it against a set of CONFLICT rules;
4. Making adjustments by relaxing constraints to have all flights assigned gates;
5. After all assignments are made, adjusting assignments to maximize gate utilization, minimize personnel workloads, maximize equipment workload.

In summary, the GATES system monitors itself in these phases: (1) First phase: uses all of the constraints; tries to schedule all the planes. With all the constraints in use, only about 75% of the planes get scheduled. However, if successful, it quits. (2) Second phase: relaxes the constraints so that all of the planes are scheduled. (3) Third phase: puts back the constraints in an attempt to optimize the schedule. All phases recurse through procedures 1, 2, and 3. Thus, procedures 1, 2, and 3 act as a kind of sub program or subprocedure which is called during every phase of the program. Phase 1 recurses only through procedures 1, 2, and 3. Phase 2 performs procedure 4, while it recurses through

1, 2, and 3. Phase 3 performs procedure 5, while it recurses through 1, 2, and 3. Because Prolog uses recursion, it's difficult to separate these into separate procedures.

Expert system—GATES. To bridge the above parallel analysis work with the ART system, we asked one developer of the system (Dr. Swigger) to make a parallel analysis of the GATES components and Shalin et al.'s and Chandrasekaran's expert system categories. This analysis is summarized in Table 8.

Table 8

A Parallel Analysis of GATES Taxonomy and Expert System Function Classification

Function Classification	GATES Function
Shalin et al.	
Classification	Classify input feature of plane type
Interpretation	Infer the schedules given data about plane type, and other descriptions
Design	Configure better Schedule using constraints of plane type, arriving and departing times
Problem-solving and Planning	Planning operators—two types of rules
Chandrasekaran	
Hierarchical Classification	No fit
Hypothesis Matching of Assessment	Process the three passes by which the system keeps refining its hypothesis and produce a better schedule
Hierarchical Design by Plan Selection and Refinement	Assign first flights and gates with the most constraints, then relax constraints to have more flights assigned
State Abstraction	No fit

Expert System Application of Human Benchmarking

Our expert system human benchmarking methodology consists of 11 steps from the initial selection of an expert system to the final report documenting the process (see Table 9). Following selection of an expert system, one would classify as to application. The possible applications are diagnosis, monitoring, planning and scheduling. Then, one would classify within a computer science taxonomy.

Table 9
Expert System Human Benchmarking Methodology

- Select expert system
- Classify as to application
- Classify within a computer science taxonomy
- Create Analogy
- Classify within a cognitive science taxonomy
- Select/develop measures of analogous functioning
- Select experimental design
- Run experimental studies with people
- Analyze statistically
- Use/create/norms
- Write report

Next, one creates an analogy, that is, the functioning of this computer software within a computer science taxonomic classification is like the specific cognitive functioning of humans. The analogous functioning is classified within a cognitive science taxonomy. Then, one selects or develops both cognitive and affective measures of human functioning on the task. Both process and outcomes are measured. Next, one selects an experimental

design and runs the experiment studies. The data is statistically analyzed and one then uses or creates norms. Finally, a report on the "intelligence" of the expert system is written.

The application of the general method to a specific case, i.e., "GATES" is shown in Table 10. GATES was selected as the expert system for reasons mentioned earlier.

Table 10

Expert System Human Benchmarking Methodology for "GATES"

General Methodology	Specific Example
• Select expert system	GATES
• Classify as to application	Scheduling
• Classify within computer science taxonomy	TBD[a]
• Create Analogy	Monitoring
• Classify within a cognitive science taxonomy	Monitoring
• Select/develop measures of analogous functioning	Thinking Questionnaire
• Select experimental design	2 X 2
• Run experimental studies with people	Completed
• Analyze statistically	TBD
• Use/create/norms	TBD
• Report "intelligence" of expert system	TBD

[a] TBD, To Be Done

It is classified as an scheduling system. Its computer science taxonomy is shown in Table 8. With respect to its analogy we consider monitoring in GATES to be like self monitoring in people (see Figure 1). For the scheduling task, the system and people have the same goal: to assign all landed flights to available gates. Both the system and people need to follow the same constraints and rules to do the task. Following these restrictions, the system monitors itself in the three phases of scheduling. People use the same constraints to plan, monitor and assess ongoing processes of scheduling. However, people are aware of

their ongoing processes while the system is not. We used an existing cognitive science taxonomy to classify monitoring processes (see Table 6). As may be seen in Figure 2, we then developed measures for both process (e.g., for people, a questionnaire on self-monitoring) and outcome. Next, an experimental design was selected in which two scheduling problems of two different difficulty levels were administered to junior college, undergraduate, graduate students, and three experts who are airport ground controllers. The experimental study has been run and the data are being analyzed. Following extensive experimental work, the creation of norms would then follow. Finally, a report on "intelligence" of the expert system will be generated. These latter steps will be the subject of an additional technical report.

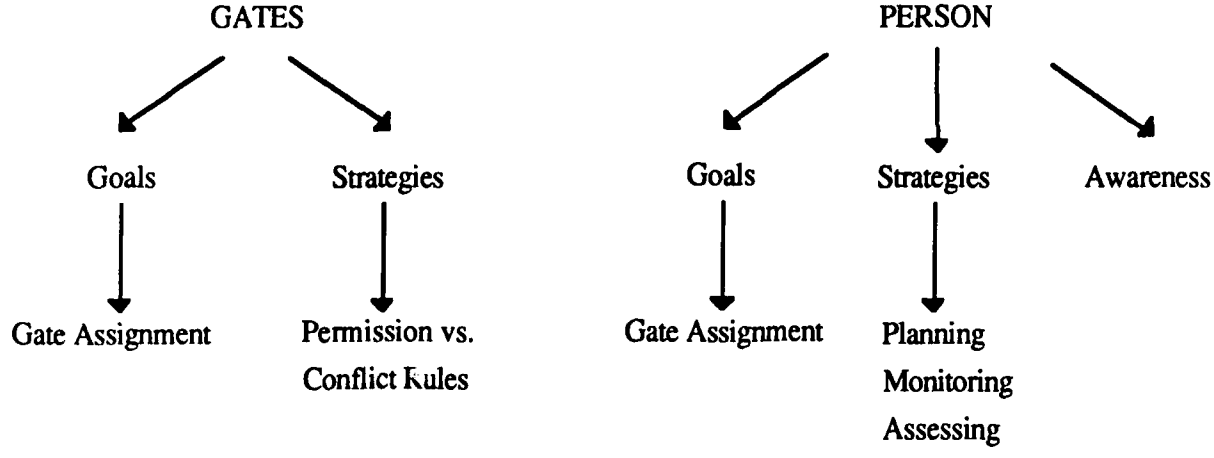


Figure 1. An analogy for human benchmarking of metacognition.

	Process	Outcome
People	a) Think-aloud protocol b) Metacognitive tests	a) Number of correct gate assignments
System	a) Trace b) Categorized rules	a) Number of correct gate assignments

Figure 2. Process and outcome measures.

Summary

This document outlined our strategy for human benchmarking of expert systems. We modified the human benchmarking methodology used for a natural language understanding system, IRUS. The metacognitive skills literature was reviewed; however, specific standardized, metacognitive tests were not found. By relating metacognitive taxonomies and expert system taxonomies, we developed a general approach to facilitate comparison. The expert system, GATES, matched our needs and was chosen as an example for our development of human benchmarking methodology for expert systems. The general methodology was developed and a specific example with GATES was instantiated. Our next report will discuss the specific design and results of an experiment in human benchmarking using GATES.

REFERENCES

Baker, E.L., & Lindheim, E. (1988, April). *A contrast between computer and human language understanding*. Paper presented at the annual meeting of the American Educational Research Association as part of the Symposium, "Understanding Natural Language Understanding," New Orleans, LA.

Baker, E.L., Turner, J.L., & Butler, F.A. (1990). *An initial inquiry into the use of human performance to evaluate artificial intelligence systems*. Center for Technology Assessment, UCLA Center for the Study of Evaluation .

Benwell, N. (Ed.). (1974). *Benchmarking: Computer evaluation and measurement.* Washington, DC: Hemisphere Publishing.

Beyer, B.K. (1988). *Developing a thinking skills program.* Boston, MA: Allyn & Bacon, Inc.

Boynton, D.A. (1986). *Planning approaches of systems experts and novices.* Dissertation, Education/Psychology Library, University of California, Los Angeles.

Brazile, R., & Swigger, K. (1988). GATES: An expert system for airlines. *IEEE Expert, 3*, 33-39

Brazile R., & Swigger, K. (1989). *Extending the GATES scheduler: Generalizing gate assignment heuristics*. Unpublished manuscript.

Brigham, M., & Pressley, M. (1988). Cognitive monitoring and strategy choice in younger and older adults. *Psychology and Aging, 3*, 249-257.

Bruner, J.S., Olver, R., & Greenfield P, (1966). *Studies in cognitive growth.* New York: Wiley.

Chandrasekaran, B. (1986). Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE Expert, 1,* 23-30.

Clements, D.H. (1987). Measurement of metacomponential processing in young children. *Psychology in the School, 24,* 23-30.

Dhar, V., & Ranganathan, N. (1988). Integer programming vs expert systems: An experimental comparison. *Communications of the ACM, 30*(3), 323-336.

Erission, K.A., & Simon, H.A. (1980). Verbal report as data. *Psychological Review, 87,* 215-251.

Flavell, J. (1981). Cognitive monitoring. In W. Dickson (Ed.), *Children's oral communication* (pp. 35-60) . New York: Academic Press.

Guralnik, D.B. (Ed.). (1984). *Webster's new world dictionary.* New York: Simon and Schuster.

Hayes-Roth, B., & Hayes-Roth, F.A. (1979). A cognitive model of planning. *Cognitive Science, 3,* 275-310.

Hayes-Roth, F., Waterman, D.A., & Lenat, D.B. (Eds.). (1983). *Building expert systems.* Reading, MA: Addison-Wesley.

Hickman, J. (1977). What do fluent readers do? *Theory into Practice, 16,* 371-375.

Hieronymus, N.A., Hoover, H.D., & Lindquist, E.F. (1986). *Iowa Tests of Basic Skills (ITBS): Teacher's guide.* Chicage, IL: Riverside Publishing.

Inference Corp. (1987). *Art.* Inference Conp., CA: Los Angeles.

Jacobs, J., & Paris, S. G. (1987). Children's metacognition about reading: Issues in definition, measurement, and instruction. *Educational Psychologist*, 22. 255-278.

Kluwe, R.H. (1987). Executive decisions and regulation of problem solving. In F.E. Weinert, & R.H. Kluwe (Eds.), *Metacognition, motivation and understanding*.(pp. 31-64). Hillsdale, NJ: Lawrence Erlbaum.

Letmanyi, H. (1984). *Assessment of techniques for evaluating computer systems for federal agency*. Final Report. Washington, DC: National Bureau of Standards (D0D), Institute for Computer Sciences and Technology.

O'Keef, R.M., Balci, O., & Smith, E.P. (1987). Validating expert system performance. *IEEE Expert*, Winter, 81-90.

O'Neil, H.F. Jr., Ni, Y., & Jacoby, A. (1990). *Literature review: Human benchmarking of expert systems*. Los Angeles: UCLA, Center for the Study of Evaluation.

Rumelhart, D.E., & Ortony, A. (1977). The representation of knowledge in memory. In R.C. Anderson, R.G. Shapiro, & W.E. Montague (Eds.), *Schooling and acquisition of knowledge* (pp. 99-136). Hillsdale, NJ: Lawrence Erlbaum.

Scholnick, E.K., & Friedman, S.L. (1987). The planning construct in the psychological literature. In S.L. Friedman & R.R. Cocking (Eds.), *Blueprints for thinking* (pp. 3-38). New York: Cambridge University Press.

Shalin, V.L., Wisniewski, E.J., & Levi, K.P. (1988). A formal analysis of machine learning systems for knowledge acquisition. *International Journal of Man-Machine Studies*, *29*, 429-446.